Deep Parametric Model for Discovering Group-cohesive Functional Brain Regions

John Boaz Lee* Xiangnan Kong* Constance M. Moore† Nesreen K. Ahmed‡

Abstract

One of the primary tasks in neuroimaging is to simplify spatiotemporal scans of the brain (*i.e.*, fMRI scans) by partitioning the voxels into a set of functional brain regions. An emerging line of research utilizes multiple fMRI scans, from a group of subjects, to calculate a single group consensus functional partition. This consensus-based approach is promising as it allows the model to improve the signal-to-noise ratio in the data. However, existing approaches are primarily non-parametric which poses problems when new samples are introduced. Furthermore, most existing approaches calculate a single partition for multiple subjects which fails to account for the functional and anatomical variability between different subjects. In this work, we study the problem of *group-cohesive functional brain region discovery* where the goal is to use information from a group of subjects to learn "group-cohesive" but individualized brain partitions for multiple fMRI scans. This problem is challenging since neuroimaging datasets are usually quite small and noisy. We introduce a novel deep parametric model based upon graph convolution, called the Brain Region Extraction Network (BREN). By treating the fMRI data as a graph, we are able to integrate information from neighboring voxels during brain region discovery which helps reduce noise for each subject. Our model is trained with a Siamese architecture to encourage partitions that are group-cohesive. Experiments on both synthetic and real-world data show the effectiveness of our proposed approach.

Keywords: functional brain analysis, fMRI, brain region discovery, deep learning, siamese neural network

1 Introduction

One of the fundamental tasks in functional analysis of the brain is the task of *functional brain region discovery*. The brain is a complex structure which is made up of various sub-structures or brain regions. The objective of functional brain region discovery is to partition voxels – from a functional Magnetic Resonance Imaging (fMRI) scan – into functionally and spatially cohesive groups (*i.e.*, brain regions). This is illustrated in Fig. 1.

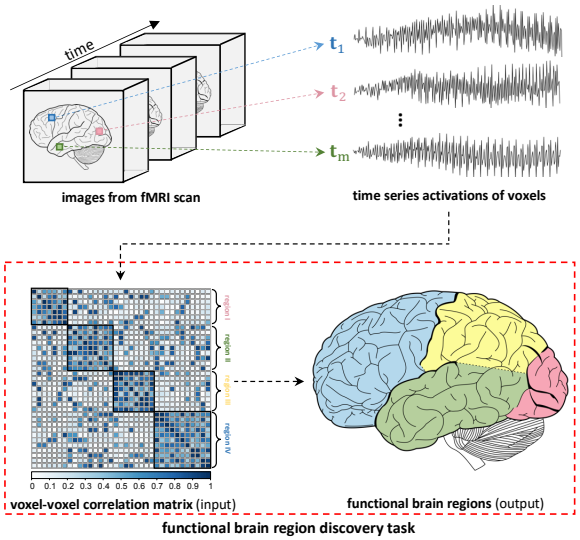


Figure 1: Functional brain region discovery aims to discover brain regions that are spatially and functionally cohesive. An example is shown here for one individual. We first take the time series activations of voxels in an fMRI scan and calculate their correlation. This is then used to recover the underlying brain regions. Due to noise, the block structures may be partially obscured making the problem challenging.

Given a set of brain regions, researchers can then analyze their relationship with each other to gain insights into the functional organization of the brain. For instance, [26] demonstrated that different regions in the brain activate when we perform different tasks.

Since the ability to find interesting and useful information during functional analysis of the brain is highly dependent on the quality of the discovered brain regions, it is important to study the problem of brain region discovery. Multiple approaches have been proposed to solve this problem [2, 9, 30].

The simplest approach relies on brain parcellation to assign voxels to established anatomical regions in a brain atlas. Here, an anatomical brain atlas (*e.g.*, AAL [30]) is used to determine which brain region each voxel belongs to. Studies such as [5, 22] use this approach to identify brain regions in brain network analysis.

However, parcellating fMRI scans using a fixed anatomical atlas may introduce noise into the data

*Worcester Polytechnic Institute, USA.
†University of Massachusetts Medical School, USA.
‡Intel Research Labs, USA.

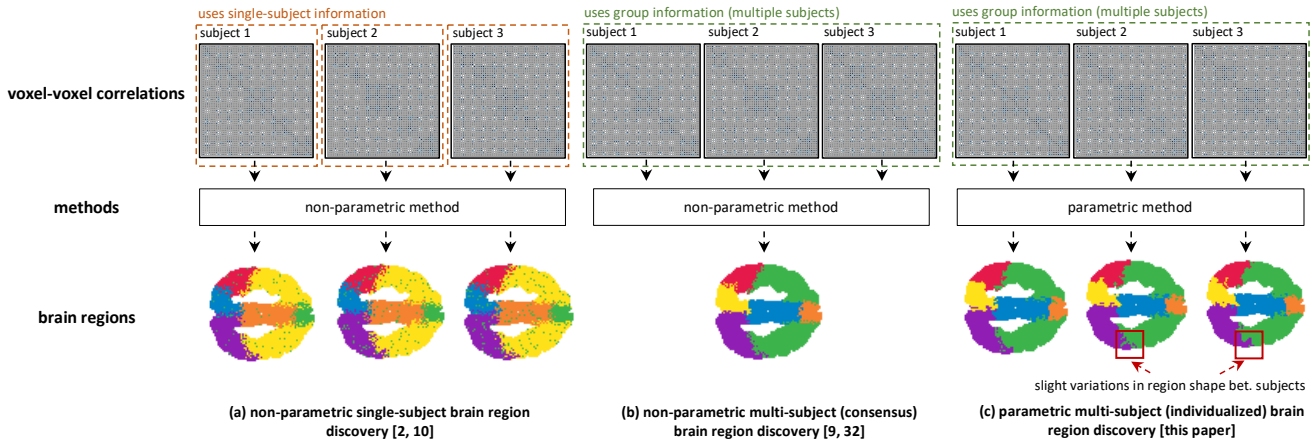


Figure 2: Different settings for functional brain region discovery. (a) Non-parametric single-subject methods [2,10] take a single fMRI scan and produce a single brain partition. When the data is noisy, the method can assign voxels incorrectly; (b) non-parametric multi-subject consensus-based methods [9, 32] take a group of subjects and produce a single consensus partition; while (c) our proposed parametric multi-subject approach produces partitions that are similar at the group-level (group-cohesion) while differing slightly per individual to account for individual variability. Furthermore, since the model is parametric, it can generalize to unseen samples.

since there is inherent variability among individuals. Just as there are variations in the shapes and sizes of human skulls, we can also expect brain regions to vary slightly across individuals [29]. Hence, multiple work [2, 9, 10, 29, 32] have been proposed instead to discover functionally cohesive regions from the data.

Among these, multi-subject consensus approaches like [9, 32] are typically more robust given the noisy nature of neuroimaging datasets [35]. To counter noise in the data, methods such as [9] and [32] derive a single brain partition whose functional regions are consistent across multiple subjects. This is in contrast to methods like [2,10] which take only a single subject at a time for functional brain region discovery.

However, these multi-subject consensus-based approaches [9, 32] risk "misclassifying" voxels lying near the boundary of regions whose boundaries shift frequently across subjects. Moreover, all the above-mentioned techniques [2,9,10,29,32] are non-parametric which means that the methods have to be re-run when new fMRI scans arrive.

In this paper we introduce a new approach called the Brain Region Extraction Network (BREN) which solves the task of *group-cohesive functional brain region discovery*. The proposed method utilizes group information (from multiple subjects) to learn "group-cohesive" but individualized brain partitions for multiple subjects. The method is able to counter noise by extracting brain regions that are consistent across multiple subjects while still capturing the small differences between individuals.

Inspired by recent work on graph convolutional networks (GCN) [13, 18], we propose a novel GCN-based approach which uses a Siamese architecture and a simple-yet-effective unsupervised loss to solve the above-mentioned task. By utilizing GCN's layer-wise propagation, our method is able to utilize information from neighboring voxels to determine which region each voxel belongs to. A Siamese architecture, on the other hand, encourages the model to discover brain regions that are group-cohesive. Fig. 2 shows the relation of our proposed approach to existing work.

2 Related Work

The discovery of functional connectivity in brains [3] has allowed us to gain insights into the functional organization of the brain. Multiple studies have shown the existence of various functional networks that emerge under various settings including those related to (1) the function of attention and eye movement [8], (2) the resting-state when the brain isn't performing an explicit task [2, 10, 17], and even (3) disease-induced states [6, 22]. The default-mode network (DMN) which becomes prominent during a person's resting-state is of particular interest as it has been shown to appear even at different levels of consciousness [16, 17].

Functional analysis of the brain usually begins with the brain network discovery problem which was first posed by [10] from a data mining perspective as the problem of simplifying spatiotemporal fMRI scans into "cohesive regions (nodes) and relationships between those regions (edges)." The two main sub-tasks are functional node/region discovery and edge discovery.

Various work have been published that attempt to solve the node discovery problem [9, 20, 32] including

that of [9, 32] which utilize a type of consensus graph cut to learn a brain partition for multiple subjects. [20] proposes a solution that calculates a single cut to partition the brain into two primary regions while our work (and that of [9, 32]) consider the more general setting of identifying an arbitrary number of regions.

Similarly, the problem of edge discovery has received much attention [5,6,22]. The work of [6] and [22] attempt to discover discriminative edges that can predict the presence of disease or neurological disorders.

A large body of work also study the problem of brain network discovery by solving both sub-tasks together [2, 34]. A notable example can be found in [2] where the problem is formulated as a matrix trifactorization with spatial regularization.

Our work is positioned among the initial group of methods which solve the functional brain region discovery task. However, we differ from existing work [9,20,32] on two main points. First, we introduce a method which produces group-cohesive *but* individualized partitions. Second, we propose to use a parametric model which can generalize well to unseen samples.

With the rise of deep learning and the success of models such as convolutional neural networks (CNN), there has been a renewed interest in studying deep architectures for graphs. Multiple work have been introduced with this goal in mind [4, 12, 18].

GCNs [18] simplify calculations by replacing principled-yet-expensive spectral graph convolutions [4] with first-order approximations. These first-order filters have been shown to work well on a variety of tasks including graph similarity [19], node classification [18], and graph classification [12]. The work that is most similar to ours is [19]. However, they tackle graph similarity while we tackle the node-level task of brain region discovery so the two are not directly comparable.

To the best of our knowledge, this is the first time GCNs have been applied to this task. Our approach is significantly different from previous work as the task of functional brain discovery is unsupervised – for which we develop a novel unsupervised loss and use a Siamese architecture to model group-cohesion. In contrast, past approaches have by and large considered tasks that fall under semi-supervised or supervised learning [14,18,33].

3 Methodology

3.1 Problem Overview We start by giving the formal definition of the problem of group-cohesive functional brain region discovery. We are given a set of M spatiotemporal fMRI scans $\mathcal{D} = \{\mathbf{S}^{(1)}, \cdots, \mathbf{S}^{(M)}\}$. Each scan, $\mathbf{S}^{(i)} \in \mathbb{R}^{D\times T}$, is comprised of D voxels each with a corresponding time-series of length T. For each scan, $\mathbf{S}^{(i)}$, we derive a corresponding non-negative affinity matrix $\mathbf{X}^{(i)} \in \mathbb{R}^{D\times D}$ – we use the absolute voxel-voxel time-series correlation matrices, in this work.

Given then, the set $\mathcal{D}' = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(M)}\}$ of affinity matrices and K which is the number of functional brain regions we wish to discover, we learn a function $f_\theta : \mathbb{R}^{D\times D} \to [0, 1]^{D\times K}$ which partitions the D voxels into K non-overlapping regions. The function f_θ, parameterized by θ, maps an input matrix $\mathbf{X}^{(i)}$ to a brain partition $\mathbf{G}^{(i)}$. The non-overlapping constraints can be ensured by imposing orthogonality between the column vectors of $\mathbf{G}^{(i)}$. That is, for $1 \le k, j \le K$, $\mathbf{g}_k^{(i)\intercal}\mathbf{g}_j^{(i)} = 0$, $\forall k \ne j$.

Under the group-cohesive setting, we wish to learn partitions that are similar across subjects to reduce the effects of noise on a single subject's fMRI scan. While $f_\theta(\mathbf{X}^{(i)}) = \mathbf{G}^{(i)}$ maps each input $\mathbf{X}^{(i)}$ to a unique partition, we want the partitions $\mathbf{G}^{(i)}$ and $\mathbf{G}^{(j)}$ to be similar, for $i \ne j$, *i.e.* $\|\mathbf{G}^{(i)} - \mathbf{G}^{(j)}\|_F^2$ should be small.

In this setting, the function f_θ is learned in an unsupervised fashion which means the labels indicating the ground-truth regions for each voxel is not provided.

3.2 Proposed Approach We begin with an introduction of the basic formulation of a GCN. For a more thorough exposition please refer to [18]. The GCN is a neural network model that is designed for graph structured data, it takes the form $f(\mathbf{X}, \mathbf{A})$ where $\mathbf{X}$ here is the input feature matrix and $\mathbf{A}$ is an adjacency matrix describing how the input nodes are related to each other – we discuss this in more detail later. The propagation rule for a general multi-layer GCN is as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}). \tag{3.1}$$

Here the superscripts l indicate the layer. Under this formulation, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of the undirected graph defined by $\mathbf{A}$ with added self loop where N is the number of nodes in $\mathbf{A}$ and $\mathbf{I}_N$ is the identity matrix of size N. Note that adding a self-loop is important because otherwise a node will not have access to its own features. The matrix $\tilde{\mathbf{D}}$, on the other hand, is defined as the diagonal degree matrix of $\tilde{\mathbf{A}}$ so, in other words, $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$. Hence, the term $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ computes a symmetric normalization for the graph defined by $\tilde{\mathbf{A}}$. Finally, $\mathbf{H}^{(l)}$ is the input to layer l of the GCN while $\mathbf{W}^{(l)}$ is the trainable weight-matrix for the same level. $\sigma(\cdot)$ here is a nonlinearity like ReLU, Sigmoid, Tanh, or Softmax [15].

It is clear from this formulation that multiple GCN layers can be chained together, much like conventional CNNs layers. In this case, we simply set $\mathbf{H}^{(1)} = \mathbf{X}$. The model is now end-to-end trainable and can be trained using stochastic gradient descent [7].

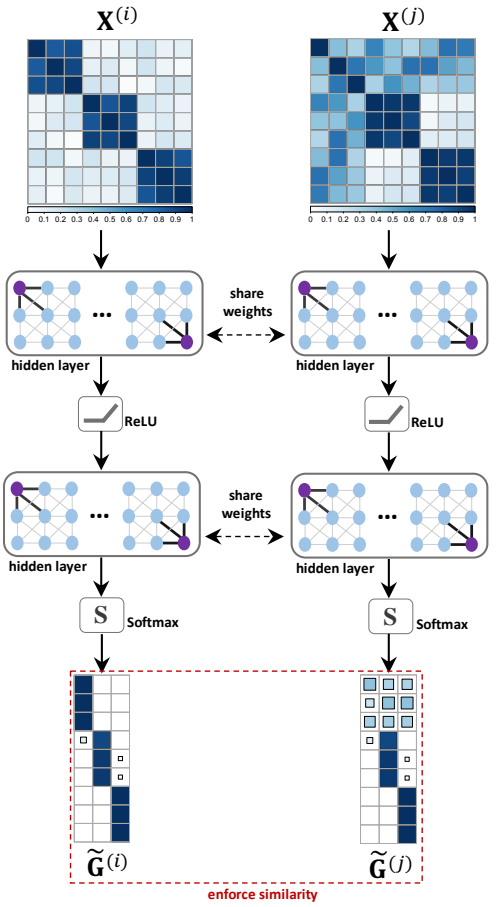


Figure 3: We show an example with $D = 9$ voxels where we are trying to discover $K = 3$ brain regions. Here, the first input $\mathbf{X}^{(i)}$ capture the correct partition well while the second input $\mathbf{X}^{(j)}$ contains noise. The Siamese architecture allows us to encourage the second partition $\tilde{\mathbf{G}}^{(j)}$ to follow that of the first subject which allows us to eventually learn similar partitions despite the noise.

3.2.1 GCN inputs In the problem overview, we kept the definition general so we stated that we learn a function $f_\theta : \mathbb{R}^{D\times D} \to \mathbb{R}^{D\times K}$ for the problem. However, since our solution is based on a GCN, we actually treat the input as graph-structured data and learn a function $f_\theta : \mathbb{R}^{D\times D} \times \mathbb{R}^{D\times D} \to \mathbb{R}^{D\times K}$. In this case, our input to the first layer of the GCN for a subject i is simply $\mathbf{H}^{(1)} = \mathbf{X}^{(i)}$ or the $D \times D$ correlation matrix for sample i. Viewed another way, each voxel j is treated as a node in a graph and its input feature is the vector $\vec{\mathbf{x}}_j^{(i)}$ describing its correlations to the other voxels in the i-th scan. Since all the scans are aligned and contain the same number of voxels, we fix the second input $\mathbf{A}$ or the adjacency matrix describing how the voxels are connected to each other.

The matrix $\mathbf{A}$ can be defined in numerous ways but in this work we simply set $A_{i,j} = 1$ when voxels i and j are vertically, horizontally, or diagonally adjacent to each other and set $A_{i,j} = 0$ otherwise. In other words, if the inputs are 2-D slices of fMRI scans, then each voxel is connected to the voxels within the 3×3 neighborhood around it and in the case of a 3-D scan, the neighborhood expands to a $3 \times 3 \times 3$ cube. By using a graph-based method like a GCN, we are able to use relevant information from a voxel's neighborhood to determine the region the voxel belongs to. This is particularly useful when we are dealing with noisy input.

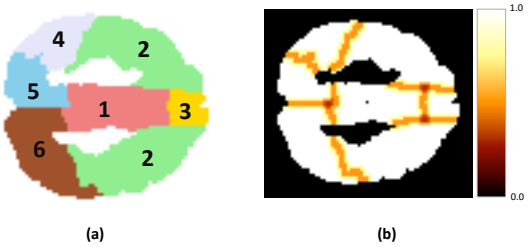
3.2.2 Design considerations Recall that given the input affinity matrix of a subject i, $\mathbf{X}^{(i)}$, the goal of the GCN is to assign each voxel to one of K discovered brain regions. To allow the GCN to perform this task, we introduce some constraints to the architecture. Given an L-layer GCN, we use a general nonlinearity (*e.g.*, ReLU, tanh, sigmoid [15, 23]) as activation for the first $L-1$ layers. For the final layer, however, we use a Softmax activation which can be viewed as a type of soft-clustering or assignment of the various voxels over the different regions. Also, we restrict the dimension of the final weight matrix $\mathbf{W}^L \in \mathbb{R}^{P\times K}$ where K is the number of regions we would like to discover and P is the feature dimension of the input to layer L.

3.2.3 Loss formulation Since the task of functional brain region discovery is unsupervised, we need a criteria that defines a good partition of the voxels without explicit guidance. As in [2], we aim to group voxels that exhibit strong functional correlation into the same group. Given an input affinity matrix $\mathbf{X}^{(i)}$ for a sample i, our L-layer GCN $f_\theta(\mathbf{X}^{(i)}, \mathbf{A}) = \tilde{\mathbf{G}}^{(i)}$ maps the input to a candidate partition $\tilde{\mathbf{G}}^{(i)}$, here $\tilde{\mathbf{G}}^{(i)} = \mathbf{H}_i^{(L+1)}$ which is simply the output of the final layer of the GCN given input $\mathbf{X}^{(i)}$. We then adjust the parameters θ of the model by minimizing the following equation:

$$\|\tilde{\mathbf{G}}^{(i)}\tilde{\mathbf{G}}^{(i)\intercal} - \mathbf{X}^{(i)}\|_F^2. \tag{3.2}$$

which can be viewed as a non-negative matrix factorization (NMF) to reconstruct the input. The process of NMF has been shown to be useful in representing objects such as the brain by learning parts that make up the whole [21]. The connection between NMF and spectral clustering approaches have been studied by [11].

To encourage the learned partitions to be more similar, we include another term in the loss formulation to explicitly model group-cohesion. For each input $\mathbf{X}^{(i)}$, we randomly select another subject's fMRI $\mathbf{X}^{(j)}$ (from the same group as i) where $i \neq j$ and we output candidate partitions for the two scans, $\tilde{\mathbf{G}}^{(i)}$ and $\tilde{\mathbf{G}}^{(j)}$ using the same network. We then adjust the parameters



(a) (b)

Figure 4: (a) Template for synthetic data with $K = 6$ regions. Region 2 has bilateral components that are spatially disjoint, mimicking certain parts of the brain [27]. (b) Probability map showing the maximal probability that a voxel belongs to a region. Note that the region assignment is uncertain along region borders to introduce inter-individual variability.

of the model θ using the updated equation:

$$\|\tilde{\mathbf{G}}^{(i)}\tilde{\mathbf{G}}^{(i)\mathsf{T}} - \mathbf{X}^{(i)}\|_F^2 + \|\tilde{\mathbf{G}}^{(j)}\tilde{\mathbf{G}}^{(j)\mathsf{T}} - \mathbf{X}^{(j)}\|_F^2 + \|\tilde{\mathbf{G}}^{(i)} - \tilde{\mathbf{G}}^{(j)}\|_F^2. \tag{3.3}$$

This can be viewed as a type of Siamese network [25]. Fig. 3 shows the design of the Siamese architecture and illustrates a case when Eq. 3.3 is particularly useful.

Optionally, similar to [2] although their usage was for penalizing far-away voxels that share a common group assignment, one can also introduce a term

$$\|\tilde{\mathbf{G}}^{(i)}\tilde{\mathbf{G}}^{(i)\mathsf{T}} - \mathbf{X}^{(i)}\|_F^2 + \|\tilde{\mathbf{G}}^{(j)}\tilde{\mathbf{G}}^{(j)\mathsf{T}} - \mathbf{X}^{(j)}\|_F^2 + \|\tilde{\mathbf{G}}^{(i)} - \tilde{\mathbf{G}}^{(j)}\|_F^2 + \beta tr\big(\tilde{\mathbf{G}}^{(i)\mathsf{T}}\mathbf{\Theta}\,\tilde{\mathbf{G}}^{(i)}\big). \tag{3.4}$$

where $\mathbf{\Theta} \in \mathbb{R}^{D\times D}$ is a matrix that encourages adjacent voxels to share the same group assignment, "tr" is the trace operator, and β is a parameter to control this spatial continuity regularization. We found, however, in our experiments that Eq. 3.3 was already sufficient in encouraging spatial cohesion. In all of our experiments, we did not find evidence to show that applying Eq. 3.4 provided any additional benefits. This seemed to show that the group-cohesion loss helped to enforce spatial continuity implicitly.

After the model is trained, we then define the final partition $\mathbf{G}^{(i)}$ for a sample i from $\tilde{\mathbf{G}}^{(i)}$ using the following step function for a hard clustering or partitioning of the voxels.

$$\begin{cases} \mathbf{G}^{(i)}_{j,k} = 1 & \max_{1\le k'\le K}\tilde{\mathbf{G}}^{(i)}_{j,k'} = k \\ \mathbf{G}^{(i)}_{j,k} = 0 & \text{otherwise.} \end{cases}$$

4 Evaluation

4.1 Region Discovery The task of identifying functional brain regions can be viewed as a group clustering problem. For our first experiment we generate synthetic data that mimic closely certain parts of the brain. Our approach is similar to that of previous work [2, 27].

Table 1: Summary of compared methods.

Method	Type	Individualized Partitions	Reference
K-Means	Single-sample	✓	[1]
Spectral	Single-sample	✓	[24]
ONMtF-SCR	Single-sample	✓	[2]
GC-I	Multi-sample	✗	[9]
GC-II	Multi-sample	✗	[32]
BREN-Basic	Multi-sample	✓	This paper
BREN-Siamese	Multi-sample	✓	This paper

For easier visualization, we follow the approach in [2] and generate synthetic data for the middle slice of an fMRI scan. The scan has dimension $91 \times 109 \approx 10,000$ voxels and we use a mask to identify voxels that belong to the brain (see Fig. 4). To generate a synthetic scan $\mathbf{S}$ (for brevity, we omit the superscripts), each of the D voxels in $\mathbf{S}$ are assigned to a group according to the probability map in Fig. 4b. For each region i, for $1 \le i \le 6$, we generate a base time-series $\mathbf{b}^{(i)}$ of length 800. Each of the time series has random mean between $[0, 10]$ and standard deviation between $[0, 2]$. The 6 base time series are generated using Cholesky Decomposition to have minimal (0.05) correlation with one another. We then use this equation to get the final signal for each voxel in $\mathbf{S}$:

$$\mathbf{s}_{i,:} = \mathbf{b}^{(\Gamma(i))} + \alpha n(i)$$

where $\Gamma : \mathbb{N} \to \{1, \cdots, 6\}$ is a function mapping each voxel to its assigned region, α is a scalar controlling the amount of noise to inject into the data, and $n(i)$ is Gaussian white noise with mean 0 and standard deviation 1. Obviously, when $\alpha = 0$, the data is well-behaved and it is trivial to partition the generated scans.

Before we delve into the analysis of results, we briefly describe the experimental setup. We run each of the compared methods ten times on randomly generated datasets. The clustering is at the voxel level. Since the cost associated with collating neuroimaging datasets is still quite high, most datasets only contain scans from a relatively few number of subjects. For instance, 26 in [32] and 21 in the healthy group of [5]. Hence, a good method should be able to generate robust results given the sparsity of training information. To mirror this challenging property of the real-world setting, we intentionally limit the number of samples $M = 20$. Finally, since we do have the ground-truth labels for the voxels in the synthetic dataset, we evaluate model performance using the Normalized Mutual Information (NMI) measure like previous work [28]. Table 1 shows a summary of the settings for various compared methods.

4.1.1 Model settings For all the compared methods, we compute the correlation between voxels and

Table 2: Avg. NMI scores (± SD) of compared methods under various noise settings.

Method	$\alpha = 1.0$	$\alpha = 1.5$	$\alpha = 2.0$	$\alpha = 2.5$
K-Means	0.687 ± 0.004	0.589 ± 0.005	0.509 ± 0.003	0.424 ± 0.003
Spectral	0.710 ± 0.002	0.626 ± 0.003	0.541 ± 0.003	0.451 ± 0.002
ONMtF-SCR	0.433 ± 0.001	0.409 ± 0.001	0.349 ± 0.001	0.241 ± 0.001
GC-I	0.775 ± 0.002	0.773 ± 0.002	0.771 ± 0.002	0.769 ± 0.002
GC-II	0.770 ± 0.003	0.759 ± 0.004	0.668 ± 0.007	0.661 ± 0.004
BREN-Basic	0.787 ± 0.011	0.780 ± 0.013	0.781 ± 0.009	0.763 ± 0.014
BREN-Siamese	**0.812 ± 0.002**	**0.797 ± 0.003**	**0.788 ± 0.002**	**0.774 ± 0.003**

apply thresholding at 0.2. For ONMtF-SCR, we use cross-validation to select the values for β and σ from $\{5, 20, 40\}$ and $\{3, 7, 10\}$, respectively. We used the author's implementation with a Python wrapper and limited max iteration to 100 as the time it took to run on 10 datasets with $M = 20$ samples was > 24 hours on a machine with 16GB of RAM and a 2.2 Ghz Intel Core i7 processor – this is already slower than other methods.

For both versions of our proposed method, we used a relatively simple architecture with $L = 3$ layers – with the number of hidden nodes set to $[75, 30, 6]$. We used a learning rate of 0.01 with the Adam optimizer and set max epoch to $2,000$ – this value was increased to $3,000$ when the noise $\alpha = 2.5$ for better convergence.

4.1.2 Quantitative analysis Table 2 shows the average NMI scores for all methods under a range of noise levels from medium ($\alpha = 1.0$) to high ($\alpha = 2.5$). We also tested all the methods under low noise settings ($\alpha = 0.2$) and outperformed the most competitive baselines (GC-I and GC-II) quantitatively as well, more analysis on this is shown in a latter portion of this paper.

Under the first case ($\alpha = 1.0$), we start to see the methods that only take a single sample deteriorate in performance. Results continue to deteriorate rapidly when the noise is increased more and more. It is particularly interesting to see that ONMtF-SCR performs pretty poorly, this may be because we limited max iterations to 100 due to speed issues. This does highlight an advantage of our method as it is parametric so results can be retrieved quickly when the model is trained.

In the case of the two group-wise methods, we see that GC-I remains fairly stable whereas GC-II starts to suffer under higher noise (2.0 & 2.5). This is quite intuitive as the averaging step in GC-I is a way to increase or improve the signal-to-noise ratio.

Our proposed method, BREN-Siamese, outperforms all compared methods under all tested noise levels. With BREN-Siamese also outperforming BREN-Basic, hinting that it is useful to enforce similarity explicitly. It is also useful to note that BREN-Siamese was found to be considerably more stable than BREN-Basic which exhibited the highest variance in performance.

Table 3: The column marked "seen" shows the performance of the model on data it was trained on while "unseen" shows the performance on data the model has never seen (*i.e.*, new unobserved fMRI scans).

Method	Setting					
	$\alpha = 0.2$		$\alpha = 1.0$		$\alpha = 2.5$	
	seen	unseen	seen	unseen	seen	unseen
BREN-Basic	0.86	0.82	0.78	0.76	0.75	0.75
BREN-Siamese	0.87	0.81	0.82	0.79	0.78	0.77

Additionally, we tested a version of our proposed method with the additional loss term (see Eq. 3.4) to encourage nearby voxels to remain in the same region but performance remained the same as BREN-Siamese which indicates that the Siamese architecture is enough without explicitly enforcing spatial continuity as in [2].

4.1.3 Visualization We now discuss some interesting things we can observe from the produced partitions. Fig. 5 shows the results of all the compared methods under low, medium, and high noise, *i.e.*, $\alpha \in \{0.2, 1.0, 2.5\}$. An interesting thing to note is that the methods that only look at one sample already struggle with noise even under minimal settings. On the other hand, we see that GC-II struggles with noise when the setting is set to $\alpha = 2.5$ while GC-I remains fairly robust.

The disadvantage of GC-I is that it produces an "average" cluster so it is unable to capture small variations across subjects. Take the case where $\alpha = 1.0$, for instance, we see that both our methods are capable of capturing the difference between the two subjects (the lower tip of region 6 colored orange and violet, respectively). This highlights another advantage of our method against methods like GC-I and GC-II.

4.2 Generalizing To Unseen Samples In previous work [2, 9, 20, 32], the proposed method was non-parametric and hence one usually had to apply the proposed method on the scans of new subjects. Since our proposed method is parametric, the trained methods can be used to partition the scans of new subjects. To verify if this is feasible, we run two tests here. In the first test, we attempt to partition the scan of new subjects whose noise levels match that of the data that was used to train the model. This is to see if there is serious degradation in performance and whether the model is overfitting. In the second test, we feed data with other noise levels to see how well a model trained using a certain level of noise can generalize.

Table 3 shows the performance of the saved models

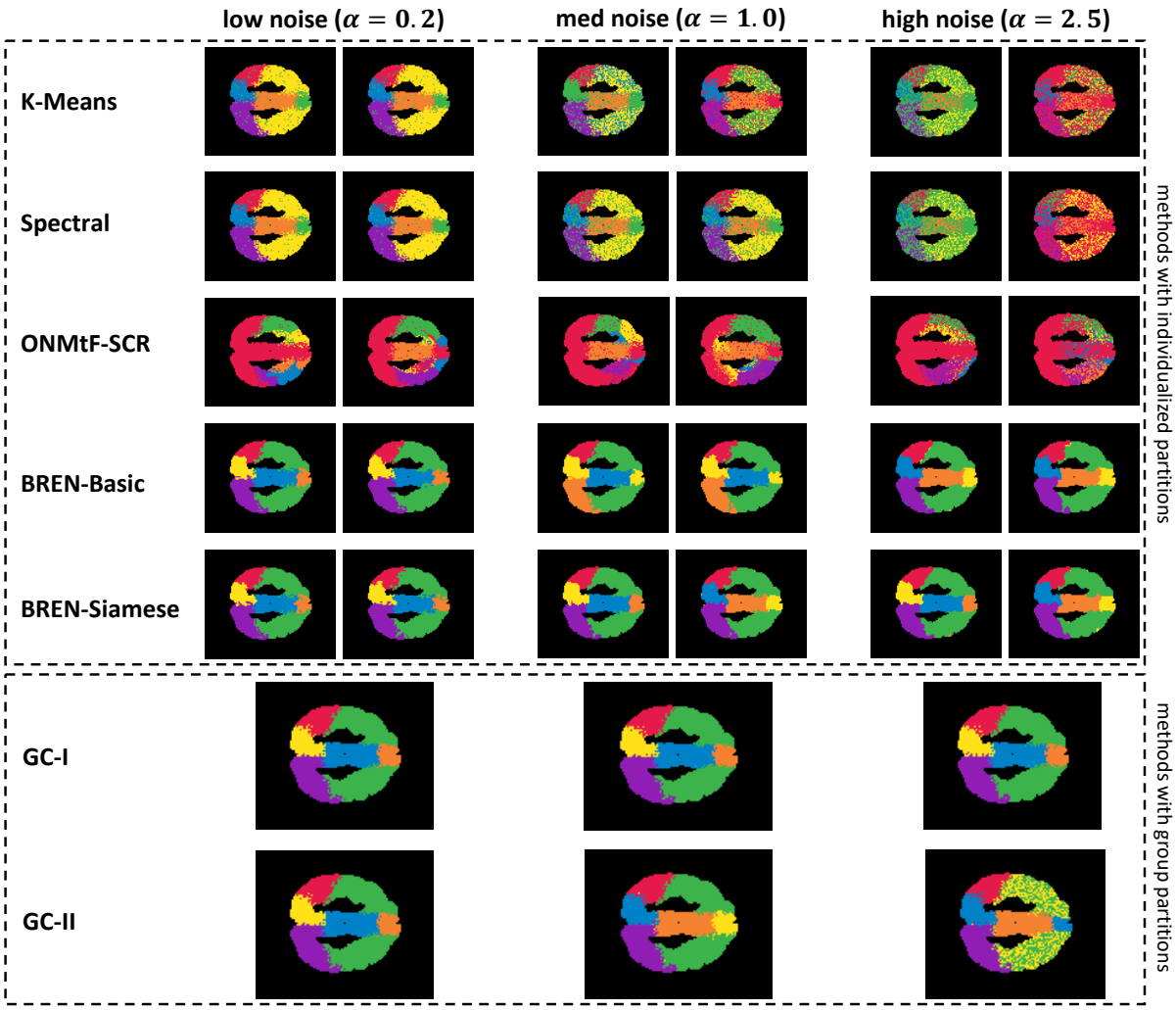
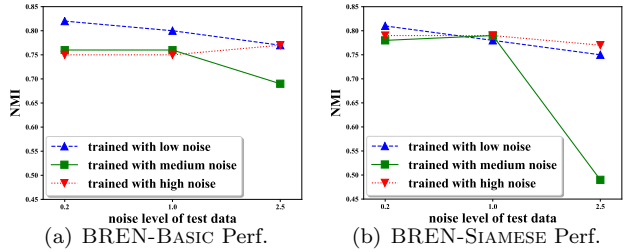


Figure 5: Resultant clusters for all compared methods under low ($\alpha = 0.2$), medium ($\alpha = 1.0$), and high ($\alpha = 2.5$) noise settings. We show two typical results for methods that produce individualized partitions.



(a) BREN-Basic Perf. (b) BREN-Siamese Perf.

Figure 6: Performance of models trained using fixed noise when tested using data with varying levels of noise.

on training (or "seen" data) and then its average performance on 50 new test (or "unseen") data generated with the same noise level. We hesitate to use the term training and test data as the task of functional brain region discovery is unsupervised. We see from the results that there is a slight drop in performance across the board which is to be expected but it is quite impressive to note that the average degradation in performance is only at 0.025 NMI which shows that the method isn't just overfitting to the training data but is learning to identify the latent regions effectively. This is very promising results as we only train each model with only 20 samples which is a fairly standard size for neuroimaging datasets.

Interestingly, we see from the table that the drop in performance is sharper on models trained with low noise ($\alpha = 0.2$) which seems to indicate that the much higher noise helps the method to generalize better much like regularization techniques [15]. In fact, we do not see a drop in performance for BREN-Basic in the last case.

Fig. 6 shows the performance of models trained *solely* on data with one of three noise levels (low, medium, and high) when tested on data with varying noise. The performance of models trained on data with low and high noise is quite stable. Their NMI scores on data with a different noise level are comparable to their scores on data with the noise level they were trained

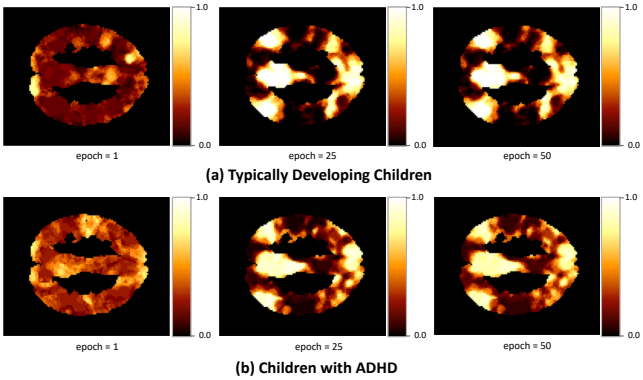


Figure 7: Average discovered DMN at epochs $\{1, 25, 50\}$ for (a) typically developing children, and (b) children suffering from ADHD.

on. Again, this may be because training on high noise is like a form of regularization. Unsurprisingly, most methods worked the best on data with the noise level they were trained on. We see, however, a sharper drop in performance of the model trained on medium noise on data with much higher noise.

4.3 Test on Real-world Dataset Finally, we test our proposed method on real-world resting-state fMRI data to demonstrate its practicality. Since the ground-truth group assignment for voxels is unavailable for real-world data, we cannot evaluate competing approaches using a quantitative measure like NMI. Instead, we follow the standard approach used by previous work [2, 10, 20] and attempt to recover the DMN.

The DMN has been shown in multiple studies [2, 10, 17] to be active when a subject is in a resting state. The voxels in the DMN exhibit high correlation with each other while being distinct from the rest of the brain. Please refer to Figs. 9 or 13 in [2] for an example.

For our experiment, we used resting-state fMRI provided by the Neuroimaging Informatics Tools and Resources Clearinghouse. In particular we used samples from the ADHD-200[1] dataset which is made accessible through the Nilearn[2] package. The dataset contains 40 fMRI scans, 20 of which belong to individuals suffering from Attention Deficit Hyperactivity Disorder (ADHD) and the remaining of which are classified as typically developing children (TDC) or adolescents.

We first take the $M = 20$ fMRI scans belonging to the TDCs and used these in our experiments, we then take the remaining scans belonging to the ADHD group and run the same experiment. Similar to [2,20], we take a single slice from each of the fMRI scans (36-th). This slice is chosen as it shows the DMN more clearly. Each slice contains 91×109 voxels, with each voxel having a corresponding time-series of length ~ 180. Like [20], we set $K = 2$ and attempt to find a partition that groups voxels into a foreground region (DMN) and a background region (rest of the brain).

We trained a BREN-SIAMESE model with $L = 3$ hidden layers with the following number of hidden nodes: $[70, 30, 2]$. This is identical to the architecture we used above except the final layer only has 2 nodes. We still used a learning rate of 0.01. This time, however, instead of feeding $\mathbf{X}^{(i)}$ (the $D \times D$ correlation matrix, for $1 \leq i \leq 20$) directly into the GCN, we used dimension-reduced data $\tilde{\mathbf{X}}^{(i)}$ instead. We used principal components analysis (PCA) [15] to reduce the dimensions of $\mathbf{X}^{(i)}$ and used this as input to speed up the training process. First, we computed the voxel-voxel correlation matrices of each of the 20 subjects, and then we used thresholding at 0.15 to remove negative and spurious correlations. We then applied PCA on the correlation matrices to reduce their dimension to 150. Hence $\tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{D \times 150}$ for all i, $1 \leq i \leq 20$.

Following the procedure employed in previous work [2, 20], we show the average networks (avg. group membership from individual scans for TDC and ADHD) that was discovered at different epochs in Fig. 7a and Fig. 7b. In both cases, we see that initially (at epoch = 1), our model cannot distinguish between the foreground (DMN) and background regions. However, after only 25 training steps, the model has already separated the voxels in the DMN from the other voxels. We use the same brain slice as [2] (36-th) and our discovered DMN closely resembles the one shown in [2] (see Figs. 9 and 13 of their paper) although we use different datasets. Our results show that the proposed method can effectively discover well-known functional regions from real-world data.

Further observation of the two discovered DMNs in Fig. 7 will show that there is less network homogeneity [31] in the average network of the ADHD group. This is highlighted in particular by the component to the right where the DMN for the ADHD group is more fragmented with darker voxel colors (this shows decreased network homogeneity). This is consistent with the findings in previous work [31] which shows that while the DMN is still prominent for people suffering from ADHD, there is usually decreased network homogeneity when compared to the scans of TDC. Note that we used the same number of scans for both groups.

5 Conclusion

In this work, we introduced a novel graph convolution method for group-cohesive functional brain region discovery. The method is able to learn group-cohesive par-

[1] http://fcon_1000.projects.nitrc.org/indi/adhd200/
[2] http://nilearn.github.io/

titions that still retain individual differences across multiple subjects. The method is also shown to be able to generalize well to unseen samples. Tests conducted on a real-world fMRI dataset show that the model can effectively discover the well-known DMN from resting-state scans for two distinct cohorts.

References

[1] C. C. Aggarwal and C. K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013.

[2] Z. Bai et al. Unsupervised network discovery for brain imaging data. In *Proc. of KDD '17*.

[3] B. Biswal, F. Z. Yetkin, V. M. Haughton, and J. S. Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic Resonance in Imaging*, 34(4):537–541, 1995.

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. In *Proc. of ICLR '14*.

[5] B. Cao et al. Mining brain networks using multiple side views for neurological disorder identification. In *Proc. of ICDM '15*.

[6] B. Cao et al. Identifying HIV-induced subgraph patterns in brain networks with side information. *Brain Informatics*, 2(4):211–223, 2015.

[7] J. Chen, J. Zhu, and L. Song. Stochastic training of graph convolutional networks with variance reduction. In *arXiv preprint arXiv:1710.10568v3*, 2018.

[8] M. Corbetta et al. A common network of functional areas for attention and eye movements. *Neuron*, 21(4):761–773, 1998.

[9] R. Craddock et al. A whole brain fMRI atlas generated via spatially constrained spectral clustering. *Human Brain Mapping*, 33(8):1914–1928, 2012.

[10] I. N. Davidson, S. Gilpin, O. Carmichael, and P. B. Walker. Network discovery via constrained tensor analysis of fMRI data. In *Proc. of KDD '13*.

[11] C. H. Q. Ding and X. He. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. of SDM '05*.

[12] D. K. Duvenaud et al. Convolutional networks on graphs for learning molecular fingerprints. In *Proc. of NeurIPS '15*.

[13] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. Protein interface prediction using graph convolutional networks. In *Proc. of NeurIPS '17*.

[14] H. Gao, Z. Wang, and S. Ji. Large-scale learnable graph convolutional networks. In *Proc. of KDD '18*.

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016.

[16] M. D. Greicius et al. Persistent default-mode network connectivity during light sedation. *Human Brain Mapping*, 29(7):839–847, 2008.

[17] M. D. Greicius, B. Krasnow, A. L. Reiss, and V. Menon. Functional connectivity in the resting brain: A network analysis of the default mode hypothesis. *Proceedings of the National Academy of Sciences of the United States of America*, 100(1):253–258, 2003.

[18] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR '17*.

[19] S. I. Ktena et al. Distance metric learning using graph convolutional networks: Application to functional brain networks. In *Proc. of MICCAI '17*.

[20] C.-T. Kuo et al. Unified and contrasting cuts in multiple graphs: Application to medical imaging segmentation. In *Proc. of KDD '15*.

[21] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(1):788–791, 1999.

[22] J. B. Lee, X. Kong, Y. Bao, and C. M. Moore. Identifying deep contrasting networks from time series data: Application to brain network analysis. In *Proc. of SDM '17*.

[23] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of ICML '10*.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. of NeurIPS '01*.

[25] Y. Qi, Y. Song, H. Zhang, and J. Liu. Sketch-based image retrieval via siamese convolutional neural network. In *Proc. of ICIP '16*.

[26] K. Rubia et al. Mapping motor inhibition: Conjunctive brain activations across different versions of go/no-go and stop tasks. *NeuroImage*, 13(2):250–261, 2001.

[27] X. Shen, F. Tokoglu, X. Papademetris, and R. T. Constable. Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *NeuroImage*, 82(1):2539–2561, 2013.

[28] Y. Sun et al. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In *Proc. of EDBT '09*.

[29] B. Thirion et al. Dealing with the shortcomings of spatial normalization: Multi-subject parcellation of fMRI datasets. *Human Brain Mapping*, 27(8):678–693, 2006.

[30] N. Tzourio-Mazoyer et al. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *NeuroImage*, 15(1):273–289, 2002.

[31] L. Q. Uddin et al. Network homogeneity reveals decreased integrity of default-mode network in ADHD. *Journal of Neuroscience Methods*, 169(1):249–254, 2008.

[32] M. van de Heuvel, R. Mandl, and H. H. Pol. Normalized cut group clustering of resting-state fMRI data. *PLOS One*, 3(4):1–11, 2008.

[33] P. Velickovic et al. Graph attention networks. In *Proc. of ICLR '19*.

[34] S. Yang et al. Structural graphical lasso for learning mouse brain connectivity. In *Proc. of KDD '15*.

[35] J. Zhang et al. Identifying connectivity patterns for brain diseases via multi-side-view guided deep architectures. In *Proc. of SDM '16*.